Assignment 9 - GPT

In this assignment, you will use various transformer models for semantic search and for language generation. We will be using the `transformers` python package from huggingface; **note** that this package will automatically download language models as required the first time the code is run, and they can be quite large. (The entire assignment might download a few GB.) You might want to do this on campus, depending on your internet situation.

This assignment is to be done individually. You may discuss the project with your classmates, but the work you turn in should be your own.

Using Generative Language Models

Goal

To learn about how generative language models can be used in practice, focusing on GPT-2, which is feasible to run locally without a graphics card.

Setup

This part uses the `transformers` package which can be installed with conda or pip.

Questions (100 pts)

1. Write a script that generates a "story" using a local GPT-2 model. Your story should: 1) be at least 100 words long; 2) not have repeated phrases; and 3) be the same every time your script is run. It might be nonsensical and/or hilarious. Use the skeleton code provided below as a starting point, and https://huggingface.co/blog/how-to-generate as a reference document.

Part 2 Deliverables

Submit your notebook as an attachment on OWL as well as a PDF version of the notebook.

Checklist

Your owl submission should include the following attachments and no additional files:

```
Assignment9.ipynb
Assignment9.pdf
```

```python
In [25]: import torch
         from IPython.display import display, Markdown
         torch_device = "cuda" if torch.cuda.is_available() else "cpu"
         from transformers import AutoTokenizer, GPT2LMHeadModel, set_seed
         tokenizer = AutoTokenizer.from_pretrained("gpt2")
         # add the EOS token as PAD token to avoid warnings
         model = GPT2LMHeadModel.from_pretrained("gpt2", pad_token_id=tokenizer.eos_token_id
```

```python
In [15]: from transformers import AutoModelForCausalLM, AutoTokenizer
         import torch

         torch_device = "cuda" if torch.cuda.is_available() else "cpu"

         tokenizer = AutoTokenizer.from_pretrained("gpt2")

         # add the EOS token as PAD token to avoid warnings
         model = AutoModelForCausalLM.from_pretrained("gpt2", pad_token_id=tokenizer.eos_tok
```

```python
In [37]: # encode context the generation is conditioned on
         model_inputs = tokenizer('Welcome to Fight Club. The first rule of Fight Club is: y

         # generate 40 new tokens
         greedy_output = model.generate(**model_inputs, max_new_tokens=40)

         print("Output:\n" + 100 * '-')
         print(tokenizer.decode(greedy_output[0], skip_special_tokens=True))
```

```
Output:
--------------------------------------------------------------------------------
----------------
Welcome to Fight Club. The first rule of Fight Club is: you do not talk about Fight
Club. The second rule of Fight Club is: you DO NOT talk about Fight Club!  You do no
t talk about Fight Club.  You do not talk about Fight Club.  You do not talk about F
ight Club.  You do not talk about Fight Club.
```

```python
In [41]: set_seed(42)

         # activate beam search and early_stopping
         sample_outputs = model.generate(
             **model_inputs,
             max_length=200,   # Targeting 100+ words
             num_beams=5,      # Beam search for diversity and fluency
             no_repeat_ngram_size=2,   # Prevent phrase repetition
             temperature=0.7,  # Add controlled randomness
             top_k=50,           # Restrict sampling to top-k tokens
             early_stopping=True,
         )
         # Decode and display the story
         story = tokenizer.decode(sample_outputs[0], skip_special_tokens=True)
```

```
In [40]: def show_decoded_tokens(dt):
             display(Markdown(dt))

         print("My GPT-2 Story:")
         print("---------------")

         ## Replace 'None' with your story; this just wraps the text
         ## to make it easier to read
         show_decoded_tokens(f"**Story:**\n\n{story}")
```

```
My GPT-2 Story:
---------------
```

Story:

Welcome to Fight Club. The first rule of Fight Club is: you do not talk about Fight Club. The second rule of Fight Club is: you DO NOT talk about Fight Club! If you want to be a part of a fight club, you have to do it in a way that makes you feel good about yourself and your team. I'm not going to go into all the details of how to get involved in Fight Clubs, but here are a few things you should know:

1. Don't be afraid to ask questions. If you don't know what you're talking about, it's probably not a good idea to start a conversation with someone who doesn't understand your point of view.
2. Be open to new ideas. There are plenty of people out there who are willing to share their ideas with you, and there are lots of other people who will be interested in your ideas, too. You can find out more about how you can join a Fight